Create Session for Reservation

The **Create Session for Reservation** endpoint enables you to export a pre-created reservation into **FrontGo** and obtain a checkout URL in a single call.

This combines reservation setup with an immediate payment session, giving the customer a seamless experience for completing the payment.

This approach is particularly useful when you already have reservation data (or have just sent a `submit reservation` request) and want to let the user pay right away without multiple round trips. The returned session can preselect a payment method (e.g., Visa, Mastercard) but still allows users to switch if needed.

Typical Flow

1. Call **Create Session for Reservation** with reservation and payment parameters.
2. Receive a `paymentUrl` and session identifiers.
3. Redirect the user to the `paymentUrl` to complete payment.
4. Receive status updates via callback as the reservation transitions through states (`Reserved`, `Captured`, `Charged`).

Prerequisites

Before integrating **Create Session for Reservation**, make sure you have the following in place:

- **Existing Reservation Data**: The endpoint assumes you have a reservation context (or that you are bundling creation & session setup).
- **API Access & Credentials**: A valid merchant account with Front Payment and access to an API environment (demo or production).
- **Bearer Token Authentication**: Include a valid Bearer token in the `Authorization` header for every request.
- **Callback Endpoint(s)**: A publicly accessible HTTPS endpoint to receive notifications (via `GET`) when status changes for the session/reservation.
- **HTTPS & JSON Support**: Your server or application must be able to send `HTTP POST` requests with JSON payloads and parse JSON responses.

- **Timestamp & Validity Control**: Ability to compute or provide valid timestamps (e.g., `dueDateForPaymentLink`) to manage how long the session remains active.
- **Testing Environment** : Access to the demo API from the following URL to validate your integration end to end before going live.

  ```
  https://demo-api.frontpayment.no
  ```

Step 1: Submit Reservation

Endpoint:

```
POST https://demo-api.frontpayment.no/api/v1/connect/reservations/create
```

Authentication

Include a **Bearer Token** in the `Authorization` header. You can obtain this token from **Front Payment**.

Example:

```
Authorization: Bearer YOUR_FRONTPAYMENT_BEARER_TOKEN
```

Request Payload Example

```
{
 "customerDetails": {
  "type": "private",
  "countryCode": "+47",
  "msisdn": "46567468",
  "email": "nafees@yopmail.com",
  "name": "Nafees",
  "preferredLanguage": "en",
  "personalNumber": null,
  "organizationId": null,
  "address": {
   "street": "Dhaka",
   "zip": "3500",
   "city": "Cumilla",
   "country": "NO"
  }
 },
```

```
  "orderDate": "1724294524",
  "dueDateForPaymentLink": "1724294524",
  "referenceNo": null,
  "customerReference": null,
  "sendOrderBy": {
    "sms": false,
    "email": false
  },
  "products": {
    "0": {
      "name": "Test",
      "productId": null,
      "rate": 1000,
      "tax": "0",
      "amount": 1000
    }
  },
  "orderSummary": {
    "subTotal": 1000.00,
    "totalTax": 0,
    "grandTotal": 1000.00
  },
  "chargeValidity": "55",
  "customerNotes": null,
  "tnc": null,
  "submitPayment": {
    "via": "visa"
  },
  "callback": {
    "callbackUrl": "https://wp.frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=success",
    "success": "https://wp.frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=success",
    "failure": "https://frontpayment.no/?order_identifier=rRbl1FWZG59o&order_status=failed"
  },
  "settings": {
    "secureDetails": false,
    "isChargePartiallyRefundable": true
  }
}
```

Validation Rules

Field	Type	Description
customerDetails.type	string	**Required**. Customer type (private or corporate).
customerDetails.countryCode	string	**Required**. Country code for the customer's phone number (e.g., "+47").
customerDetails.msisdn	string	**Required**. Mobile phone number without country code.
customerDetails.email	email	**Required**. Valid customer email address.
customerDetails.name	string	**Required**. Full name of the customer.
customerDetails.preferredLanguage	string	**Required**. Customer preferred language. Available languages are en, no, sv, da, de. If nothing is given it will set default to no.
customerDetails.personalNumber	string	**Optional**. Customer's personal identification number, must be exactly 11 characters containing only numbers and cannot contain spaces. When Customer type is private then you can used this for add personal number.
customerDetails.organizationId	string	**Required** Organization identification number, must contain only numbers and cannot contain spaces. When Customer type is corporate then this field is required. Otherwise you can add this as null or remove from payload.
customerDetails.address	array	**Required**. Customer address details.
customerDetails.address.street	string	**Required**. Street name.
customerDetails.address.zip	string	**Required**. Postal code.
customerDetails.address.city	string	**Required**. City name.
customerDetails.address.country	string	**Required**. ISO Alpha-2 country code (e.g., NO). Custom validation IsoAlpha2Country applies..
orderDate	string	**Required**.Unix timestamp for the Date of the order.
dueDateForPaymentLink	string	**Required**.Provide Current / Future Date as Unix timestamp for the Due Date of the order.
referenceNo	string	**Optional**. Reference number.
customerReference	string	**Optional**. Customer reference ID.
sendOrderBy	array	**Required**. Notification preferences.
sendOrderBy.sms	boolean	**Required**. Whether to send order by SMS.
sendOrderBy.email	boolean	**Required**. Whether to send order by email.
products	array	**Required**. List of product items.
products.*.name	string	**Required**. Name of the product..

Field	Type	Description
products.*.productId	string	**Optional**. Unique identifier for the product.
products.*.rate	numeric	**Required**. Rate per unit of the product.
products.*.tax	numeric	**Required**. Tax rate must be (e.g., 0, 12, 15, 25), Unless you have other configuration.
products.*.amount	numeric	**Required**. Total product amount.
orderSummary.subTotal	numeric	**Required**. Subtotal of all products before tax and discount.
orderSummary.totalTax	numeric	**Required**. Total tax for the order.
orderSummary.grandTotal	numeric	**Required**. Grand total of the order.
customerNotes	string	**Optional**. Notes from customer.
tnc	string	**Optional**. Terms and conditions.
chargeValidity	string	**Optional**. Validity in minutes (digits only).
submitPayment	array	**Required**. Payment submission details.
submitPayment.via	string	**Required**. Payment method (visa , mastercard).
callback	array	**Optional**. Callback URLs.
callback.callbackUrl	url	**Optional**. General callback URL.
callback.success	url	**Optional**. Success redirect URL.
callback.failure	url	**Optional**. Failure redirect URL.
settings	array	**Optional**. Additional settings.
settings.secureDetails	boolean	**Optional** If secureDetails is True , order details wouldn't be visible without verifying BankID.
settings.isChargePartiallyRefundable	boolean	**Optional**. If the value is true charge will be party refundable, otherwise it will be fully refundable.

Example Success Response


```
{
  "status_code": 201,
  "status_message": "OK",
  "message": "Reservation Submitted Successfully",
  "is_data": true,
  "data": {
    "customerUuid": "CSRT3463048878",
    "reservationUuid": "RES4161996022",
    "paymentUrl": "https://v1.checkout.bambora.com/aa7ec3f47b0d45b286bcc595ab0d9613"
```

```
  }
}
```

Error Response

401 Unauthorized: Missing or invalid Bearer token. Other validation errors will return appropriate HTTP error codes (e.g., 400 Bad Request) along with error messages specifying the invalid or missing fields.

```
{
   "status_code": 500,
   "status_message": "Internal Dependency Error",
   "message": "internalErrorOccurredPleaseTryAgainLater",
   "is_error": true,
   "errors": {
      "happenedAt": "String",
      "internalErrorDetails": "Array"
   }
}
```

```
{
   "status_code": 510,
   "status_message": "Internal Dependency Error",
   "message": "Payment Gateway Error: Failed to Create Checkout Session",
   "is_error": true,
   "errors": [
      "Payment Gateway Error(Submit Payment): Failed to Create Checkout Session At: PAR156"
   ]
}
```

Step 2: Payment Process

- From the success response in **Step 1**, the user is redirected to the `paymentUrl`.
- The preselected payment method (`visa` or `mastercard`) will be shown, but the user can change it.
- After successful payment:
  - The third-party system is notified via the provided `callbackUrl`.
  - The user is redirected to the `success` or `failure` URL provided in the request payload.

Notifications via Callback URL

The `callbackUrl` is an endpoint on your server that our system will call via an `HTTP GET` request whenever the status of the specified order changes from its initial state. For `reservation` the `callbackUrl` will be triggered for three status changed: `Reserved`, `Captured`, `Charged`.

See the link below to understand how to work with the callback URL on your side and how to verify the request sent from our side.

Go To `Notication Via Callback Url` Page

Revision #36
Created 22 January 2024 10:08:03 by Admin
Updated 30 September 2025 12:36:23 by Nayamot Ullah